[0094] CLAIMS

What is claimed is:

1. A computer system comprising:

a memory; and

a processor that supports SIMD instructions, the processor being configured to perform Montgomery multiplication using SIMD instructions.

2. A computer system as recited in claim 1, wherein the processor is executing a cryptographic function and the Montgomery multiplication is used to compute exponentiations in the cryptographic function.

3. A computer system as recited in claim 1, wherein the processor maintains two arrays to hold intermediate computations from the Montgomery multiplication, and the SIMD instructions are used to simultaneously update the two arrays.

4. A computer system as recited in claim 1, wherein the Montgomery multiplication involves a first multiplication of an input array and a second multiplication of a modulus array, and the SIMD instructions are used to perform simultaneously the first and second multiplications.

5. A computer system as recited in claim 1, wherein the Montgomery multiplication has a loop of instructions, and each iteration of the loop involves, excepting copy operations, using no more than eight SIMD instructions.

6. A computer system as recited in claim 5, wherein the SIMD instructions comprise two load instructions, one multiply instruction, two add instructions, one copy instruction, one bitwise AND instruction, one store instruction, and one shift instruction.

7. A processing system comprising:

a processor having a set of registers, the processor being configured to support SIMD instructions; and

a set of SIMD instructions, executable by the processor, to perform Montgomery multiplication:

$$\text{montmul}(A, B) = \text{rem}(\,(AB - qN)/R, N), \quad \text{where} \quad q = \text{rem}(AB\,N', R).$$

where A and B are integers, q is a quotient, N is a modulus, R is an integer that is coprime to modulus N, and N' is an integer such that $N\,N' \equiv 1 \pmod{R}$.

8. A processing system as recited in claim 7, wherein the SIMD instructions comprise a single SIMD instruction that simultaneously performs parts of the multiplications AB and qN.

9. A processing system as recited in claim 7, wherein the integer B and the modulus N are implemented as arrays, and at least one SIMD instruction is used to update a first array T_1 with multiples of B for computing AB and to update a second array T_2 with multiples of N for computing qN.

10. A processing system as recited in claim 9, wherein a single SIMD instruction is used to update the first array T_1 and the second array T_2 simultaneously.

11. A system as recited in claim 9, wherein:

a first register holds elements of the B and N arrays;

a second register holds an element of the first array T_1 and an element of the second array T_2; and

a third register is used to hold results of the first array T_1 being updated with a multiple of B and the second array T_2 being updated with multiples of N.

12. A computer readable medium comprising computer-executable SIMD instructions that, when executed, direct a processor to perform Montgomery multiplication.

13. A computer readable medium as recited in claim 12, comprising:

a first SIMD instruction to load elements of arrays B and N into a first register;

a second SIMD instruction to load elements of arrays T_1 and T_2 into a second register;

a third SIMD instruction to multiply an element in the array B by a first multiple and an element in the array N by a second multiple;

fourth and fifth SIMD instructions to add results of the third SIMD instruction to the array elements loaded by the second SIMD instruction and to any carries saved from a previous iteration;

sixth and seventh SIMD instructions to separate each output of the fifth SIMD instruction into two reduced size results, one that fits into the arrays T_1 and T_2 and another that represents a carry for a next iteration; and

an eighth SIMD instruction to update an element of array T_1 and an element of array T_2, in memory.

14. A computer readable medium as recited in claim 12, wherein the SIMD instructions comprise SSE2 instructions.

15. A method for computing Montgomery multiplication:

$$\text{montmul}(A, B) = \text{rem}(\ (AB - qN)/R, N), \quad \text{where} \quad q = \text{rem}(AB\,N', R).$$

where A and B are integers, q is a quotient, N is a modulus, R is an integer that is coprime to modulus N, and N' is an integer such that $N\,N' \equiv 1 \pmod{R}$, the method comprising:

iteratively performing, for each digit of integer A from right to left:

with array T_1 being updated by a product of input B times the digit of integer A, determining what multiple of modulus N allows the updated arrays T_1, T_2 to end with the same digit;

multiplying the input B by the digit of integer A and multiplying the modulus N by the determined multiple; and

updating the arrays T_1, T_2.

16. A method as recited in claim 15, wherein the performing comprises using SIMD instructions.

17. A method as recited in claim 15, wherein the multiplying is performed by a single SIMD instruction.

18. A method as recited in claim 15, further comprising initializing the arrays T_1, T_2 and the modulus N prior to said performing.

19. One or more computer readable media storing computer executable instructions that, when executed, perform the method as recited in claim 15.

20. A method comprising:

initializing a set of registers with values used in performing Montgomery multiplication; and

computing the Montgomery multiplication with SIMD instructions on the values stored in the registers.

21. A method as recited in claim 20, wherein the computing comprises using the Montgomery multiplication to compute exponentiations in a cryptographic function.

22. A method as recited in claim 20, wherein the computing comprises using SSE2 instructions.

23. A method as recited in claim 20, wherein the Montgomery multiplication has a loop of instructions, and each iteration of the loop is performed using not more than nine SIMD instructions.

24. A method as recited in claim 23, wherein the nine SIMD instructions comprise two load instructions, one multiply instruction, two add instructions, one copy instruction, one bitwise AND instruction, one store instruction, and one shift instruction.